Articulated and Generalized Gaussian Kernel Correlation for Human Pose Estimation

Meng Ding, Student Member, IEEE, and Guoliang Fan, Senior Member, IEEE

Abstract—In this paper, we propose an articulated and generalized Gaussian kernel correlation (GKC)-based framework for human pose estimation. We first derive a unified GKC representation that generalizes previous sum of Gaussians (SoG)-based methods for the similarity measure between a template and an observation both of which are represented by various SoG variants. Then we develop articulated GKC (AGKC) by integrating a kinematic skeleton in a multivariate SoG template that supports subject-specific shape modeling and articulated pose estimation for both the full body and hands. We further propose a sequential (body/hand) pose tracking algorithm by incorporating three regularization terms in the AGKC function, including visibility, intersection penalty and pose continuity. Our tracking algorithm is simple yet effective and computationally efficient. We evaluate our algorithm on two benchmark depth datasets. The experimental results are promising and competitive when compared with state-of-the-art algorithms.

Index Terms—Kernel Correlation, Sum of Gaussians (SoG), Articulated Pose Estimation, Human Pose Tracking, Hand Pose Tracking, Shape Modeling, Depth Sensor, Kinect.

I. INTRODUCTION

Articulated human/hand pose estimation is one of the fundamental research topics in the field of computer vision and machine learning due to its wide applications and related technologies, such as Human Computer Interaction (HCI), Robotics, Computer Animation and Biomechanics. Over the past few decades, color image-based human/hand motion estimation and analysis have been intensely researched, and hundreds of studies can be found in the reviews [1]–[3]. Recently, the launch of low-cost RGB-D sensors (e.g., Kinect) has further triggered a large amount of research due to the additional depth information and easy foreground/background segmentation. The existing algorithms can be roughly categorized into three groups, i.e., discriminative, generative and hybrid. The approaches in the first group are usually efficient and may require a large database for querying or training [4], [5]. Those in the second group involve an articulated body model for template matching [6], [7] which is often computationally costly and requires a good initialization and sequential tracking for efficient implementation. Those in the third category are intended to take advantage of both ideas [8]–[11].

To capture human motion efficiently from multi-view 2D images, a shape model based on the sum of Gaussians (SoG) (i.e., the univariate SoG) was developed in [12]. This simple yet effective shape representation provides a (nearly) differentiable model-to-image similarity function, allowing fast pose estimation. SoG was also used in [13]–[15] for both human and hand pose estimation. In our early work [16], a generalized SoG model (GSoG) (i.e., the multivariate SoG) was proposed, where it encapsulated fewer anisotropic Gaussians for human shape modeling, and a similarity function between GSoG and SoG was defined in the 3D space. In a similar spirit, a sum of anisotropic Gaussians (SAG) model was developed in [17] for hand pose estimation, where the similarity is measured by the projected overlap in 2D images. Both GSoG and SAG have improved the performance of pose estimation compared with the original SoG methods. In this work, we provide a unified framework that generalizes all above approaches from the perspective of Kernel Correlation-based registration [18]. Specifically, we extend the Gaussian kernel correlation (GKC) from the univariate to the multivariate case and derive a general similarity function between two collections of arbitrary Gaussian kernels. We also embed a kinematic skeleton into the Gaussian kernels, leading to a tree-structured articulated GKC (AGKC) controlled by a group of quaternion-based rotations. Given the input point set represented by Gaussian kernels, pose parameters can be estimated by maximizing the AGKC between the template and the input data, as shown in Fig. 1, where our framework is presented for pose estimation of the full body and a hand.
Our unified framework is able to handle any pairwise comparison, including $\text{SoG} \leftrightarrow \text{SoG}$, $\text{SoG} \leftrightarrow \text{GSoG}$, $\text{GSoG} \leftrightarrow \text{GSoG}$, and even $(\text{SoG}+\text{GSoG}) \leftrightarrow (\text{SoG}+\text{GSoG})$. The last two new cases offer great flexibility and generality for articulated registration. There are mainly three contributions in this work.

- First, we develop a generalized Gaussian kernel correlation function from the univariate case to the multivariate case in $n$ dimensional space, along with a unified and differentiable similarity measure between any $\text{SoG}$ and $\text{GSoG}$ combinations.

- Second, we present an articulated kernel correlation function for shape modeling and pose estimation where the tree-structured template is represented by a few multivariate Gaussian kernels along with quaternion-based rotations.

- Third, by introducing three regularization terms (visibility, continuity and self-intersection), we propose an efficient and robust sequential pose tracking algorithm, which is successfully applied to pose estimation both body and hand from a single depth sensor.

Our algorithm is simple and efficient and can run at about 10 FPS on a i7 desktop PC without GPU acceleration. We evaluate our articulated pose tracking algorithm on two depth benchmark datasets, i.e., [8] (body) and [19] (hand), which shows that the accuracy of pose estimation is competitive compared to the best results reported so far [6], [10], [20]. The rest of the paper is organized as follows: First, we briefly review some related work in Section II. Then, the generalized Gaussian kernel correlation (GKC) is presented in detail in Section III. Articulated GKC (AGKC) is presented in Section IV. We present the sequential pose tracking algorithm in Section V. The experimental results are shown in Section VI, followed by the conclusion in Section VII.

II. RELATED WORK

We review related works from three perspectives, i.e., *kernel correlation for registration, body/hand pose estimation, and articulated shape modeling*, all of which are related to our work as shown in Fig. 2.

A. Kernel Correlation (KC) for Registration

According to how the template and the target are matched, registration approaches can be classified into two major categories, i.e., correspondence-based and correspondence-free. The algorithms in the first category iteratively estimate the correspondences and the underlying transformation, such as the Iterative Closest Point (ICP) [21] and the Maximum Likelihood-based density estimation [22]–[25]. The algorithms in the second group directly optimize an energy function without involving correspondences, including density alignment [26] and kernel correlation [18]. Different from the density alignment whose energy function is a discrepancy measure using $L_2$ distance, kernel correlation was first presented as a similarity measure in [27] and it was used for point set registration in [18], where both the template and the scene are modeled by kernels and their registration is achieved by maximizing a KC-based similarity measure. KC was also applied to the stereo vision-based modeling in [28]. When the kernel function is a Gaussian, there are two unique benefits for registration, i.e., robustness and efficient optimization. First, as stated in [26], GKC in rigid registration is equivalent to the robust $L_2$ distance between two Gaussian mixture models (GMMs). Similarly, it was stated in [28] that GKC is equivalent to a distance measure between two data sets in the M-estimator [29]. Second, different from the Maximum Likelihood-based registration using Expectation-Maximization (EM) [23]–[25], GKC supports a direct gradient-based optimization that is more efficient and robust. However, existing GKC mainly considers the case of univariate (isotropic) Gaussian with two exceptions (to the best of our knowledge). First, SoG was extended to sum of anisotropic Gaussians (SAG) in [17] where the similarity function was evaluated in the projected 2D image space. Our previous work [16] studied anisotropic Gaussians in 3D space and derived a similarity measure between the template and target, represented by multivariate and univariate Gaussians, respectively. In this work, we generalize both approaches by developing the $n$-dimensional Gaussian KC function that supports a unified similarity measure between two collections of arbitrary univariate/multivariate Gaussian kernels.

B. Human/Hand Pose Estimation

As mentioned before, the approaches to pose estimation from a depth sensor can be roughly categorized into three groups. First, discriminative methods extract depth features, like [5], [30], and then reconstruct a pose by either searching in a database or directly predicting the location of body/hand joints. In [5], [31], [32], a random forest classifier was trained from a large dataset to label depth pixels as body/hand parts. A sufficiently large training database is necessary for
C. Articulated Shape Modeling

A good articulated shape model is essential which not only captures shape variability among different individuals but also facilitates pose estimation with robustness and accuracy. One of the most widely used shape models is the mesh surface [6], [37], [38] that is able to be deformed smoothly for articulate pose estimation. GPU-based implementation is often necessary for real-time processing. Some other methods use a collection of geometric primitives, like spheres and cylinders to render the object surface that is compared to the observed shape cues for matching [7], [11], [36], [39], [40]. For example, in [7], a geometric representation was used to estimate the human pose by an improved ICP. On the other hand, the parametric shape representation becomes popular [12], [14]–[17], [41]–[43]. In particular, a SoG-based parametric shape model was developed in [12] and it is amendable for articulated shape modeling and pose estimation. Compared with the mesh surface and geometric primitives, parametric models are simpler with a lower computational load. It is worth noting that the geometric shape models and parametric ones are closely related but different in the way the models are involved in the cost function during optimization. In this paper, we develop a new articulated KC function for parametric shape representation that is composed of a collection of multivariate/univariate Gaussian connected by a kinematic skeleton.

III. GENERALIZED GAUSSIAN KERNEL CORRELATION

In this section, we generalize the original Gaussian kernel correlation in [18] from two aspects. First, we extend the univariate Gaussian to the multivariate one and derive a unified GKC function between two Gaussians in n dimensional space. Second, we provide a more general kernel correlation between two collections of Gaussian kernels, both of which can be composed by univariate/multivariate Gaussian kernels (Fig. 4 (a-c)) or even the mixed kernel model (Fig. 4 (d)).

A. A Unified Gaussian Kernel Correlation

Given two Gaussians centered at points \( \mu_1, \mu_2 \in \mathbb{R}^n \), their kernel correlation is defined as the integral of the product of two Gaussian kernels over the n dimensional space [18],

\[
KC(\mu_1, \mu_2) = \int_{\mathbb{R}^n} G(x, \mu_1) \cdot G'(x, \mu_2) dx,
\]

(1)

where \( x \in \mathbb{R}^n \), and \( G(x, \mu_1), G'(x, \mu_2) \) represent the Gaussian kernels centered at the data point \( \mu_1, \mu_2 \), respectively. Different from [18], where the Gaussian kernel has a standard univariate Gaussian distribution form, we employ an non-normalized Gaussian kernel defined in [44],

\[
G'(x, \mu) = \exp\left(-\frac{||x - \mu||^2}{2\sigma^2}\right),
\]

(2)

where the superscript "(u)" represents "univariate" and \( \sigma^2 \) is the variance. The non-normalized Gaussian kernel can lead to a more controllable and meaningful kernel correlation between two Gaussian with large differences in variance, because the non-normalized \( G \) and \( G' \) have a similar scale even if their variances \( \sigma_1, \sigma_2 \) are largely distinct, as shown in Fig. 3.

![Fig. 3. The comparison of normalized (left) and non-normalized (right) Gaussian kernels with the same variances \( \sigma_1, \sigma_2 \).](image)

Plugging (2) in (1), it is straightforward to have the kernel correlation of two (non-normalized) univariate Gaussian at \( \mu_1 \) and \( \mu_2 \),

\[
UKC(\mu_1, \mu_2) = \left(2\pi \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}\right)^{\frac{n}{2}} \exp\left(-\frac{||\mu_1 - \mu_2||^2}{2(\sigma_1^2 + \sigma_2^2)}\right).
\]

(3)

If the variance \( \sigma^2 \) is extended to the covariance matrix \( \Sigma \), we have the non-normalized multivariate Gaussian kernel form,

\[
G'(x, \mu) = \exp\left(-\frac{1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu)\right).
\]

(4)

Obviously, when \( \Sigma \) is a diagonal matrix and the diagonal entries are identical, (4) will degenerate to (3). Now, we re-write (1) using (4) to derive the generalized Gaussian kernel correlation, which is not as straightforward as (3). The derivation details can be found in Appendix A. Finally, we have the kernel correlation of two n dimensional multivariate Gaussian kernels which are centered at points \( \mu_1, \mu_2 \) and modeled by the covariance matrices \( \Sigma_1, \Sigma_2 \) respectively,

\[
MKC(\mu_1, \mu_2) = \sqrt{\frac{(2\pi)^n}{|\Sigma_1 + \Sigma_2|}} \exp\left(-\frac{1}{2}(\mu_1 - \mu_2)^T (\Sigma_1 + \Sigma_2)^{-1} (\mu_1 - \mu_2)\right).
\]

(5)

Different from statistical correlation to represent the proximity of two distributions in the statistics, our kernel correlation,
where the non-normalized Gaussian kernels are involved, is defined as a kind of energy to measure the similarity of two parametrical models. In other words, the energy becomes larger as the two kernel models become closer and more similar to each other.

B. KC of Two Collections of Gaussian Kernels

Several Gaussian kernels that are centered at a set of points \( \Omega = \{ \mu_1, \ldots, \mu_m \} \) can be combined as a sum of Gaussian kernels \( \mathcal{X} \),

\[
\mathcal{X} = \sum_{i=1}^{m} G(\mathbf{x}, \mu_i).
\]

Given two collections of Gaussian kernels \( \mathcal{X}_A \) and \( \mathcal{X}_B \) composed by \( M \) and \( N \) Gaussian kernels respectively, their kernel correlation is defined as,

\[
MKC(\mathcal{X}_A, \mathcal{X}_B) = \int_{\mathbb{R}^n} \sum_{i=1}^{M} \sum_{j=1}^{N} G(\mathbf{x}, \mu_{i})G(\mathbf{x}, \mu_{j})d\mathbf{x} \\
= \sum_{i=1}^{M} \sum_{j=1}^{N} MKC(\mu_{i}, \mu_{j}).
\]

where \( MKC(\mu_{i}, \mu_{j}) \) has been derived in (5). It is worth noting that \( \mathcal{X}_A \) and \( \mathcal{X}_B \) can be composed of univariate Gaussians (Fig. 4 (a)), multivariate Gaussians (Fig. 4 (b,c)) or mixed Gaussians (Fig. 4 (d)). Consequently, we obtain a unified kernel correlation function in (7) to evaluate the similarity between any pairwise combination of univariate and multivariate SoG models, as shown in Fig. 4. When the covariance matrices in \( \mathcal{X}_B \) degenerate to variances in the 3D space, the degenerated equation (7) will be equivalent to the SoG-GSoG similarity in [16]. Further, if the covariance matrices in \( \mathcal{X}_A \) degenerate to variances in 3D, (7) will become the SoG-GSoG similarity in [13]–[15]. Both degenerations imply that our kernel correlation functions in (5) and (7) generalize all the previous SoG-based methods.

IV. ARTICULATED KERNEL CORRELATION

In this section, we first embed an articulated skeleton in a collection of Gaussian kernels where quaternion-based 3D rotations are involved to represent the transformation between two segments along the skeleton. Then, based on the generalized KC in (7), a segment-scaled articulated Gaussian kernel correlation is proposed to balance the effect of each segment in the articulated structure.
corresponding body joint marked as red stars in Fig. 5 (a), the index \( l \) is used in both the body joint and its associated segment. In this work, each joint in the body has 3 degrees of freedom (DoF) rotation, and the joints marked with the red circles and stars in the hand model (Fig. 5 (b)) have 1 DoF and 3 DoF rotation, respectively. If \( l \) is the root joint (the hip joint), \( T_{\text{root}} \) is the global transformation of the whole body. Given a transformation matrix \( T_j \), the center of \( j \text{th} \) Gaussian kernel in the segment \( S_j \) at the T-pose \( \bar{\mu}_{j,k} \) can be transferred to its corresponding position in the world coordination, \[ \bar{\mu}_{j,k} = T_j \bar{\mu}_{j,k}. \] (9)

Accordingly, the local transformation \( R \) at each joint and \( T_{\text{root}} \) defines a specific pose. Since the translation between two segments is pre-defined, only rotation is to be estimated in each \( R \). In this work, we express a 3D joint rotation as a normalized quaternion due to that can facilitate gradient-based optimization. Here, we have \( L \) joints (\( L = 10 \), marked as red stars in Fig. 5 (a)), each of which allows a 3 DoF rotation represented by a quaternion vector of four elements. Also, there is a global translation at the hip (root) joint. As a result, we have a total of 43 parameters in a full body pose represented by \( \Theta \). In the hand model, since 1 DoF rotation is controlled by two elements of a quaternion, there are totally 47 parameters. Similar to (9), given the T-pose model \( \bar{\mathcal{K}}_A \), the deformed one under pose \( \Theta \) is,
\[
\mathcal{K}_A = \mathcal{K}_A(\Theta) = \sum_{l=1}^{M} G(x, \bar{\mu}_l^{[A]}(\Theta)). \tag{10}
\]

Consequently, the Gaussian kernels are embedded into an articulated skeleton and controlled by the quaternion-based pose variable \( \Theta \). This articulated Gaussian kernel-based shape representation is general and can be applied to any other articulated shape models. Re-writing (7) using (10), we explicitly obtain the articulated Gaussian kernel correlation as,
\[
MKC(\mathcal{K}_A(\Theta), \mathcal{K}_B) = \sum_{l=1}^{M} \sum_{j=1}^{N} MKC(\bar{\mu}_l^{(A)}(\Theta), \bar{\mu}_j^{(B)}), \tag{11}
\]

where \( MKC(\bar{\mu}_l^{(A)}(\Theta), \bar{\mu}_j^{(B)}) \) can be calculated in (5). As a similarity measure, the analytical representation of our articulated kernel correlation in (11) become the main part of our objective function. As a result, the problem of articulated pose estimation becomes to finding the optimal \( \Theta \) by which the deformed template \( \mathcal{K}_A(\Theta) \) has the maximum kernel correlation with \( \mathcal{K}_B \), i.e., Gaussian Kernel-based representation of an observed point cloud. Next, we further propose a new segment-scaled Gaussian kernel correlation to balance the effect of each segment in an articulated structure.

### B. Segment-scaled Gaussian Kernel Correlation

The Gaussian kernel correlation \( MKC(\mathcal{K}_A(\Theta), \mathcal{K}_B) \) can be evaluated according to (11) and (5). In practice, we found that the kernel correlation from larger segments (e.g. torso in the human body or palm in the hand) could dominate the energy function, overshadowing contributions from small segments. For example, we show the GKC defined in (11) for five body segments in the first 50 frames of Sequence 17 in Fig. 7 (a). It is obvious that the GKC value of torso is much larger than those from other segments. This bias may trap the optimizer in a wrong local minimum, since the gradient direction is also mostly affected by the large segments.

To balance the contributions from different segments in the holistic kernel correlation, we further upgrade (11) to balance the influence of each articulated segment, referred to as “segment-scaled Kernel Correlation”. Specifically, the kernel correlation from body segment \( S_j \) is weighted by a coefficient \( \frac{1}{\omega_l} \) as,
\[
sMKC(\mathcal{K}_A(\Theta), \mathcal{K}_B) = \sum_{l=1}^{L} \frac{1}{\omega_l} \sum_{k=1}^{K_l} \sum_{j=1}^{N} MKC(\bar{\mu}_{l,k}^{(A)}(\Theta), \bar{\mu}_j^{(B)}), \tag{12}
\]

where \( K_l \) is the number of Gaussian kernels in the segment \( S_j \) (in total, we have \( L \) segments with the equality \( K_1 + \cdots + K_l + \cdots + K_L = M \)), and \( \frac{1}{\omega_l} \) means the weight of the corresponding segment \( S_j \). Without loss of generality, we calculate \( \omega_l \) as the integral of all the Gaussian kernels in the segment \( S_j \),
\[
\omega_l = \int_{\mathbb{R}^n} \sum_{k=1}^{K_l} G(x, \bar{\mu}_{l,k}) dx = \frac{K_l}{\int_{\mathbb{R}^n} |\Sigma_k|^{1/2}} \tag{13}
\]

where \( \omega_l \) denotes the volumetric measure of segment \( S_j \). In other words, the larger body segment, the greater the value of \( \omega_l \) (i.e., the smaller the weight). Fig. 7 (b) shows the segment-scaled GKC of five segments, which are much more comparable after scaling. It is worth mentioning that \( \omega_l \) is calculated during shape learning and used for online tracking.
V. PROPOSED POSE TRACKING ALGORITHM

In this section, we first propose a subject-specific shape modeling method. Then, we introduce the objective function for pose tracking with three additional regularization terms, followed by a fast gradient-based optimization algorithm. Moreover, we develop a failure detection and recovery strategy to ensure robust and smooth sequential pose tracking.

A. Subject-specific Shape Modeling

We develop an efficient two-step approach to estimate the subject-specific shape model that is represented by a multivariate SoG along with a certain-sized skeleton. To simplify the optimization process, we first use an auxiliary SoG-based template that consists of 57 univariate Gaussian kernels for skeleton/shape learning, and then we convert it to the final shape model composed of 13 multivariate Gaussian kernels that is suitable for articulated pose tracking. This approach effectively reduces the space of SoG parameters and still takes advantage of the multivariate SoG for shape modeling.

![Fig. 8.](image)

In the first step, we choose a fully-stretched initial pose to support accurate estimation of the bone lengths and body shape for each new subject, as shown in Fig. 8. We want to loosen the rigid body constraints and to allow free movement of each Gaussian kernel for better adapting to the observation under a “neutral” pose. A set of SoG parameters (in total $57 \times 4 = 228$), $\mathbf{\Pi}$, which defines the location and variance of each univariate Gaussian is optimized by maximizing the KC function defined in (12). However, some Gaussian kernels from different body parts could be blended near joints, as shown in Fig. 8 (b). To avoid this problem, we augment a Local Linear Embedding (LLE)-based topology constraint [45], which aims to preserve the articulated structure in the auxiliary SoG-based shape representation. The new objective function for the subject-specific shape modeling is defined as:

$$
\hat{\mathbf{\Pi}} = \arg \min_{\mathbf{\Pi}} \left\{ -UKC(\mathbf{\hat{X}}(\mathbf{\Pi}), \mathbf{X}_B) + \lambda \sum_{i=1}^{M} ||\mathbf{\mu}_i - \sum_{j \in \tau_i} w_{ij} \mathbf{\mu}_j||^2 \right\},
$$

where $\mathbf{\mu}_i$ is the mean of the $i$th Gaussian in the body model; $\tau_i$ represents the $K$ nearest neighbors ($K = 4$ in this work) of the $i$th Gaussian; $w_{ij}$ is the LLE weight; $\lambda$ controls the weight of the LLE term. Large $K$ could limit the flexibility of each Gaussian kernel to match the subject-specific shape; small $K$ may not provide sufficient topology constraint to preserve the articulated body structure. This objective function can be optimized by an nonlinear optimizer, like [46]. The subject-specific SoG-based body model is shown in Fig. 8 (c), where all Gaussian kernels are re-distributed to better fit the observed subject in in Fig. 8 (a) while keeping their original relative positions.

In the second step, we map the univariate SoG to the multivariate SoG model through a pre-defined multiple-to-one mapping relationship. For example, six univariate Gaussian kernels at the top-left part of the torso in the auxiliary SoG model are mapped to a multivariate Gaussian kernel in the multivariate SoG model as shown in Fig. 8 (c) and (d). First, we compute the mean of the multivariate Gaussian kernel by averaging the means of six univariate Gaussian kernels that usually have similar variances. Then we use PCA of six Gaussian means to find the three principal components and associated eigenvalues which are used construct the covariance matrix of the multivariate Gaussian kernel. Due to the flatness of depth data, there is a very small eigenvalue, and thus we reset it to be the averaged variance of the six univariate Gaussians to have better volumetric representation. The estimated subject-specific shape model is shown in Fig. 8 (d). This two-step shape learning method can also be used in articulated hand modeling.

B. Objective Function of Pose Tracking

The goal of the pose tracking algorithm is to estimate the pose parameters $\Theta$ at time $t$ from an observed point cloud by minimizing an objective function and utilizing previous pose information. The framework is shown in Fig. 9. We define our objective function that includes the articulated Gaussian kernel correlation $sMKC(\mathbf{X}_A(\Theta), \mathbf{X}_B)$ defined in (12) along with three additional regularization terms. The first is a visibility detection term $Vis$ to cope with the incomplete data problem from self-occlusion; the second one is a new intersection penalty $E_{int}(\Theta)$ to discourage the intersection of two body segments; the third one is a continuity term $E_{con}(\Theta)$ to enforce a smooth pose transition during sequential tracking. Then pose estimation is formulated as an optimization problem with the following objective function:

$$
\hat{\Theta} = \arg \min_{\Theta} \left\{ -\sum_{l=1}^{L} \sum_{g=1}^{K_l} \sum_{k=1}^{N} MKC(\mathbf{\mu}^{(A)}_{l,k}(\Theta), \mathbf{\mu}^{(B)}_{j}) \right\} - \eta E_{int}(\Theta) + \gamma E_{con}(\Theta),
$$

where the first term is the negative of $sMKC$ in (12); $E_{int}(\Theta)$ and $E_{con}(\Theta)$ are the intersection and continuity term respectively; $\lambda, \gamma$ are the weights to balance the last two terms, and $Vis(l,k)$ is the visibility of the $k$th Gaussian in the segment $S_l$, defined as,

$$
Vis(l,k) = \begin{cases} 
0 & \text{if the Gaussian is invisible,} \\
1 & \text{otherwise.}
\end{cases}
$$

In the following, we introduce each term in details.

1) Kernel Correlation Term: The AGKC term is defined in (12) and (5). It is noted that maximizing the kernel correlation function is equivalent to minimizing its negative. To use AGKC, the observed point cloud should also be represented
by a SoG-based model. In this work, instead of a Quad-tree used in [12] to cluster the image pixels with a similar color, we employ an Octree to directly partition the point cloud which is very efficient to down-sample a point cloud while preserving 3D spatial information. Octree clustering is also robust to outliers and noise by removing relatively small clusters. In the Octree partitioning, if points in a Octree node has a large standard deviation along the depth direction (greater than a threshold $\eta_{\text{depth}}$), we divide the node into eight sub-nodes, up to a maximum Octree level $n_{\text{level}}$. Then, points in each leaf node cube (illustrated as adjacent points in the same color in Fig. 10 (b)) are represented by an isotropic (univariate) Gaussian $G_j$ centered at the mean of the points with the variance $s_j^2$ that is set to be the square of half-length of a side of the cube. Consequently, we obtain a compact and noise-reduced univariate SoG representation $\mathcal{K}_B$ of a point cloud as shown in Fig. 10 (c).

2) Visibility Detection Term: To address the incomplete data problem like Fig. 11 (a), we develop a visibility detection term to identify and exclude the invisible Gaussian kernels from the subject shape model. Similar to [47], the pose in the previous frame is used to detect the visibility. Our idea is that a large overlap among multiple Gaussians in the projected image plane may indicate an occlusion. To compute the overlap area analytically, we again use the auxiliary univariate SoG (the one used in the first-step shape learning) for occlusion handling. First, each Gaussian of the template model under the previous pose is orthographically projected to the 2D image plane along the depth direction, resulting in a set of circles whose radii are set to be the square root of the corresponding variances.

Then, we compute the overlap area between every two circles. As shown in Fig. 11 (c), if the overlap area of any pairwise circles is larger than certain percentage (e.g. $\frac{1}{3}$) of the area of the smaller one, we declare an occlusion. The Gaussian kernel which is closer to the camera is remained, otherwise, it is occluded. Then, we map the auxiliary SoG model to the multivariate SoG model with the pre-defined mapping, which has been used for shape modeling in Section V-A. Finally, we count the number of occluded circles in each body segment to decide its visibility. If more than half kernels in a body segment are invisible, the corresponding segment is excluded during optimization.

3) Intersection Penalty Term: In previous SoG-based methods [13]–[15], to avoid the situation that two or more body segments intersect with each other, an artificial clamping function was used to constrain the energy contribution of each Gaussian kernel in $\mathcal{K}_B$. However, this clamping operation introduces some discontinuity to the objective function, which may hinder the performance of the gradient-based optimizer. In this paper, we develop an intersection penalty term to replace the artificial clamping function that is naturally deduced from the proposed GKC framework in (11). The idea is that two separated segments are treated as template $\mathcal{K}_a$ and target $\mathcal{K}_b$, and then their KC is used to measure their intersection as:

$$E_{\text{int}}'(\Theta) = MKC(\widehat{\mathcal{K}_a}(\Theta), \mathcal{K}_b).$$

When two segments intersect each other, their KC becomes large, resulting in a larger intersection penalty. In practice, we consider five self-intersection cases, i.e., head-torso, forearm-arm, upper limb-torso, shank-thigh and lower limb-torso.
$E_{int}(\Theta)$ that is the sum of KC measures of the five cases can be considered as a soft constraint which preserves the continuity and differentiability of the objective function.

4) Continuity Term: To encourage smooth sequential tracking, we introduce a continuity term as follows,

$$E_{\text{con}}(\Theta^{(t)}) = \sum_{d=1}^{D} \left[ \left( \Theta^{(t)}_d - \Theta^{(t-1)}_d \right) - \left( \Theta^{(t-1)}_d - \Theta^{(t-2)}_d \right) \right]^2,$$

(18)

where $\Theta^{(t)}$ is the present pose and $\Theta^{(t-1)}, \Theta^{(t-2)}$ are the previous two poses; $d$ represents the dimension index in $\Theta$. The continuity term penalizes the current pose to have a large deviation from previous frames, ensuring relatively smooth pose transition over time.

C. Gradient-based Optimization

Due to the differentiable AGKC function and the computational benefits of quaternion-based rotation representation, we can explicitly derive the derivative of the objective function $E$ with respect to $\Theta$ and employ a gradient-based optimizer. Different from a variant of steepest descent used in [12], [13], we employ a Quasi-Newton method (L-BFGS [46]) because of its faster convergence. For simplicity, we ignore the visibility detection term in (15) and have the following form:

$$\frac{\partial E(\Theta)}{\partial \Theta} = -\frac{\partial sMKC(\tilde{X}_A(\Theta), X_b)}{\partial \Theta} + \lambda \frac{\partial E_{int}(\Theta)}{\partial \Theta} + \gamma \frac{\partial E_{\text{con}}(\Theta)}{\partial \Theta}$$

$$= -\frac{1}{N} \sum_{l=1}^{L} \sum_{k=1}^{K_l} \sum_{j=1}^{N} MKC(\mu^{(l)}_j, \mu^{(l)}_j) \frac{\partial}{\partial \Theta}$$

$$+ \lambda \frac{\partial E_{int}(\Theta)}{\partial \Theta} + \gamma \frac{\partial E_{\text{con}}(\Theta)}{\partial \Theta}. \quad (19)$$

We denote $r = [r_1, r_2, r_3, r_4]^T$ as an un-normalized quaternion, which is normalized to $p = [x, y, z, w]^T$ according to $p = \frac{r}{\|r\|}$. We represent the pose $\Theta$ as $[t^{(t-1)}, \ldots, t^{(l)}]$, where $t = [t_1, t_2, t_3] \in \mathbb{R}^3$ defines a global translation, $L$ is the number of joints to be estimated, and each normalized quaternion $p^{(l)}$ from $r^{(l)} \in \mathbb{R}^4$ defines the relative rotation of the $l_{th}$ joint. Defined in (5), $\mu_{t,k} = [a, b, c]^T$ is the center of $k_{th}$ Gaussian kernel in the segment $S_l$ which is transformed from its local coordinate $\mu_{l,k}$ through transformation $T_l$ in (9) and the corresponding covariance matrix $\Sigma_{l,k}$ is approximated and updated from the previous pose under an assumption that adjacent poses should be close to each other. We explicitly represent every pairwise kernel correlation using (5) and take derivative with respect to each pose parameter, which will be summed over to obtain the gradient vector of our kernel correlation:

$$\frac{\partial MKC}{\partial t_n} = \frac{\partial MKC}{\partial \mu_{l,k}} \frac{\partial \mu_{l,k}}{\partial t_n}, \quad (n = 1, 2, 3) \quad (20)$$

$$\frac{\partial MKC}{\partial t^{(j)}_{m}} = \frac{\partial MKC}{\partial \mu_{l,k}} \frac{\partial \mu_{l,k}}{\partial t^{(j)}_{m}} \frac{\partial t^{(j)}_{m}}{\partial T_l} \frac{\partial T_l}{\partial r^{(j)}_{m}}, \quad (m = 1, \ldots, 4) \quad (21)$$

which are straightforward to calculate. The derivative of $E_{int}(\Theta)$ can also be calculated by a similar way according to (20), (21). Since $E_{\text{con}}(\Theta^{(t)})$ in (18) is a standard quadratic form, we have its gradient expression directly as:

$$\frac{\partial E_{\text{con}}(\Theta^{(t)})}{\partial \Theta^{(t)}} = 2 \left[ \left( \Theta^{(t)}_d - \Theta^{(t-1)}_d \right) - \left( \Theta^{(t-1)}_d - \Theta^{(t-2)}_d \right) \right],$$

(22)

where $d = 1, \ldots, D$. The initialization of $\Theta^{(t)}$ is the estimated pose in the previous frame and the pose in the first frame is assumed to be close to the standard T-pose, similar to the treatment in many other algorithms.

D. Failure Detection and Recovery

Although gradient-based local optimization is effective in most cases, it is still possible to be stuck at local minima and not be able to recover automatically, especially when there is a dramatic and fast pose change or significant self-occlusion. To cope with this problem, we incorporate Particle Swarm Optimization (PSO) with gradient-based search to balance the effectiveness and efficiency when exploring the high-dimensional parameter space [48]–[50]. To reduce the computational load, some data-driven detectors will be helpful to provide a good initialization and narrow the search space. In [19], some finger detectors are used to effectively combine gradient-based ICP and sampling-based PSO for real-time articulated hand tracking. Similar ideas can be incorporated in our tracking framework where Gaussian KC-based optimization is treated as the local optimizer and PSO is used for global search. Additional detectors are necessary to support real-time performance of the hybrid global-local optimization that are beyond the scope of this work.

The hybrid optimization with PSO and AGKC is only necessary when a tracking failure is detected. We evaluate the average KC for all $N$ univariate Gaussian kernels in the observation $(X_b)$ by checking the following condition:

$$\frac{1}{N} sMKC(\widetilde{X}_A(\Theta), X_b) < \eta_{\text{fail}}, \quad (23)$$

where $sMKC(\cdot)$ is defined in (12). When (23) is true, it indicates that a number of Gaussian kernels in $X_b$ are not aligned or explained by the deformed shape template $(\widetilde{X}_A)$. Then the local-global optimization scheme will be triggered for failure recovery when PSO is involved to allow the global PSO sampling along with the local gradient-based AGKC optimization.

VI. EXPERIMENTAL RESULTS

A. Experiment Setup

Testing Database: We first use the depth benchmark dataset SMMC-10 [8] to evaluate our algorithm for human pose tracking by comparing with state-of-the-art methods. The SMMC-10 dataset consists of 28 depth sequences, which include various human motion types. The ground truth data are the 3D marker positions that are recorded by an optical tracker. The significant noise and outliers in this depth dataset make it challenging yet proper for evaluating algorithm robustness and accuracy. Second, we also use the benchmark dataset in [19] to test our algorithm for hand tracking. This dataset is reported as
one of the most challenging ones due to the fast hand motion and significant self-occlusion. Performance evaluation on the first dataset is both quantitative and qualitative to validate the efficacy of our algorithm for human pose tracking, while that of the second one is mainly qualitative to demonstrate the potential of the proposed framework for a different articulated structure.

**Evaluation Metrics:** We adopt two metrics for performance evaluation of human pose estimation. One evaluation metric is to directly measure the averaged error of the Euclidean distance between the ground-truth markers and estimated ones over all markers across all frames,

$$\bar{e} = \frac{1}{N_f} \sum_{k=1}^{N_m} \frac{1}{N_f} \sum_{i=1}^{N_m} \| \mathbf{p}_{ki} - \mathbf{v}_{i}^{\text{disp}} - \hat{\mathbf{p}}_{ki} \|,$$

(24)

where $N_f$ and $N_m$ are the number of frames and markers; $\mathbf{p}_{ki}$ and $\hat{\mathbf{p}}_{ki}$ are the ground-truth location of the $i_{th}$ marker and the estimated one in the $k_{th}$ frame, respectively; $\mathbf{v}_{i}^{\text{disp}}$ is the displacement vector of the $i_{th}$ marker. Because the marker definitions across different body models are different, the inherent and constant displacement $\mathbf{v}^{\text{disp}}$ should be subtracted from the error, as a routine in most methods. In this paper, we manually chose 40 frames with ground truth in the #6 sequence for the calculation of $\mathbf{v}^{\text{disp}}$. To make $\mathbf{v}^{\text{disp}}$ independent of any pose, we project each marker on the centerline of its corresponding segment and compute an offset $\mathbf{v}^{\text{disp}}$ in the local coordinate system for each segment individually. The other evaluation metric is the percentage of correctly estimated joints whose Euclidean distance errors are less than 10cm.

**Algorithm Parameters:** Some empirical parameters we used throughout our experiments are listed. In Octree partitioning, the threshold $\eta_{\text{depth}}$ and maximum Octree level $m_{\text{level}}$ are set to be 20mm and 6, respectively. The weights $\eta$ and $\gamma$ in (15), and $\lambda$ in (14) are set to be 0.2, 0.001 and 0.05, respectively.

### B. Effect of the Additional Terms

To exhibit the effect of each regularization term introduced in the objective function, we conduct five experiments on the SMMC-10 dataset, where the continuity, visibility detection and intersection penalty terms as well as the subject-specific shape model are incorporated successively. Their corresponding tracking errors are shown in Fig. 12 (a), which shows that the tracking accuracy gradually improves with the addition of each of the three terms as well as the subject-specific shape model. Especially, in Sequences 24-27 where the occlusion problem is serious, the visibility and intersection terms make a significant contribution. It is also interesting to find that the continuity term has a a slight negative effect in Sequence 25 (Karate) due to its too strong penalty on the fast motion. However, the other terms and the shape model are able to improve the accuracy. Fig. 12 (b) and (c) illustrate the tracking error of the left elbow in Sequence 24 and that of the left knee in Sequence 27 respectively. It is clear that using additional terms (in red) achieves much smaller errors than the case without them (in blue). We visually compare the effect of the additional terms in Fig. 13, where it is observed that the results using additional terms (in green) are more accurate.

### C. Accuracy Comparison

In Fig. 14 and Fig. 15, our algorithm is evaluated against the state-of-the-art methods in terms of two metrics. Failure recovery is only needed for Sequences 24, 25 and 27, and our approach achieves the average error 3.56cm on the SMMC-10 dataset and it is close to the best results so far (around 3.4 ~ 3.6cm) [6], [10], [20] where a database or a detailed mesh model are involved. If no failure detection and recovery
are involved with real-time performance for all sequences, the average error is 3.71 cm. As shown in Fig. 14, our method achieves the best result in Sequences 0-23 where the human motion is relatively smooth with little occlusion. On the other hand, our results are a little worse than the best ones in Sequences 24-27, since the simplicity of our shape model makes it hard to handle large non-rigid body deformation and occlusion problems in complex motions. Nevertheless, our correlation-based (correspondence free) registration approach is computationally more efficient and still provides comparable precision of joint estimation (Metric II) with the best algorithms [6], [7] as shown in Fig. 15. Moreover, we notice that our results are better than the original SoG algorithm (reported in [15]) and [47] where additional inertial sensors were used. It also outperforms our early GSoG method [16], which is mainly due to the proposed segment-scaled AGKC and the continuous intersection penalty term.

D. Efficiency Analysis

For mesh-based generative methods, the computational complexity is expressed as $O(MN)$, where $M$ is the number of vertices in a surface model and $N$ is the number of points in the observation point set. In our experiment, due to the effective down-sampling of Octree, $N$ is about 300-500, which is much less than that in other methods. Due to the multivariate SoG body shape representation, $M$ in our approach is much less than those in most methods and $M$ in the multivariate SoG is only about a quarter of that in the standard SoG, leading to a low computational cost. We implement our tracking algorithm in C++ with the L-BFGS optimization library [51]. Currently, the efficiency is evaluated on a PC without GPU acceleration. We allow a maximum of 30 iterations in the first frame (similar to a standard T-pose) and then 15 iterations in the following frames, and we ignore the computation time of background segmentation using a depth threshold and the efficient Octree partitioning. We can achieve about 20 frames per second without the code optimization for human pose tracking. If the hybrid local-global optimizer is employed in three sequences (#24, #25, #27), the computational cost is increased due to PSO-based failure recovery, leading to a lower frame rate. In this work, we used 10 particles and 20 generations in the PSO-assisted local-global optimizer to test the effectiveness of the failure detection and recovery. However, it is possible to keep the real-time performance if our algorithm can be integrated with some data-driven detectors as those used in [19] to initialize and reduce the search space. Due to the collective nature of AGKC and PSO, our algorithm (with failure recovery) is compatible with GPU-based parallel computing for fast implementation.

E. Effect of Failure Detection and Recovery

We track the average AGKC value in each frame according to (23) to detect a failure. As mentioned earlier, only three SMMC-10 sequences (#24, #25 and #27) have a couple of detected failures. However, most hand sequences require failure recovery due to fast and complex articulated hand motion. Fig. 16 shows the average AGKC with/without the failure recovery in sequence #25 of SMMC-10 and sequence...
#1 of hand motion. As shown in Fig. 16 (a) and (b), pose estimation fails from frame #174, where its average AGKC value drops below the threshold ($\eta_{fail} = 9$). Then, the recovery is triggered in the following frames, until the average AGKC value becomes larger than $\eta_{fail}$. Without failure recovery, the pose tracker could be trapped in local minima in the following frames, as shown in the red curve in Fig 16 (b). On the other hand, Fig 16 (c) visualizes the recovered pose estimation result in frame 200. Similar results for a hand sequence are shown in Fig 16 (d,e,f), where the failure is detected in frame 74 and a good recovery is obtained at frame #97. While most tracking failures can be successfully recovered for full-body pose tracking, the current hybrid optimization strategy is still not ready to handle complicated hand motion yet. The main reason is that AGKC has too many local minima in hand tracking, which deteriorates when there are fast articulated pose changes and complex self-occlusion problems. A more advanced failure detector [52] could be helpful to reduce false alarms. More importantly, some finger detectors similar to that used in [19] could mitigate this problem by reducing the search space and providing a better optimization initialization.

**F. More Discussion**

Some pose estimation results of SMMC-10 sequences are shown in Fig. 17. While the estimated poses are accurate in most frames for all sequences, and the failure recovery is only triggered in a couple of frames in three sequences, our tracker may still fail in a few frames of some sequences, as shown in the last row of Fig. 17. We also evaluate our algorithm on several sequences from the hand dataset and compare with the ground truth qualitatively in Fig. 19. Since the hand motion is rapidly changing and highly articulated, there exists significant self-occlusion in most hand sequences. Failure detection and recovery are required for most hand sequences. Although the hybrid optimizer shows promising results in our experiments, it may still fail in some frames of highly complex articulated motion. Some hand tracking failures are shown in Fig. 18.

There are two possible reasons that will guide our future research. First, the visibility term in the objective function may not be accurate since it is determined from the previous frame, especially in the case of fast motion or view angle change. We could address this by incorporating the predicted pose into the visibility term or allowing the visibility term to be optimized. Second, there are still many local minima in the objective function mainly due to the self-occlusion problems, and a better optimizer is needed to take advantage of the differentiability of AGKC. PSO is effective but costly, and it must be confined to a small search space. Integrating an additional pose detector or some bottom-up features could improve initialization and narrow the search space, which are the two main keys to efficient and effective optimization in articulated pose tracking of the full-body and hands.
VII. CONCLUSION

We have developed a generalized Gaussian KC (GKC) framework that provides a continuous and differentiable similarity measure between a template and an observation, both of which are represented by a collection of univariate and/or multivariate Gaussians. We further develop an articulated Gaussian KC (AGKC) function by embedding a quaternion-based articulated skeleton in a multivariate SoG model. Consequently, pose parameters are estimated by maximizing AGKC along with three additional constraints. Also, the new AGKC function naturally supports a differentiable intersection term to discourage the overlap between body segments, which is better than the artificial clamping function used before. We have evaluated our proposed tracker on two public depth datasets, and the experimental results are encouraging and promising compared with the state-of-the-art algorithms, especially considering its simplicity and efficiency. It may be possible to introduce other tree structure (e.g., KD tree) to further improve the efficiency of AGKC optimization by focusing on those kernel pairs that are spatially close. Our algorithm can achieve fast and accurate human pose estimation with competitive accuracy and precision, and the proposed GKC and AGKC functions can also be applied to other articulated structures.

ACKNOWLEDGEMENTS

The authors would like to thank the reviewers for their comments and suggestions that improved this paper.

APPENDIX

PROOF OF THE UNIFIED GAUSSIAN KERNEL CORRELATION EQUATION

The proof of the unified Gaussian kernel correlation in (5) is listed below. Given two non-normalized Gaussian kernels centered at two points $\mu_1, \mu_2$, we aim to derive their kernel correlation $KC_m(\mu_1, \mu_2)$ which is represented as,

$$MKC(\mu_1, \mu_2) = \int_{\mathbb{R}^n} G_1^{(m)}(x, \mu_1) \cdot G_2^{(m)}(x, \mu_2) dx.$$

We re-write $G_1^{(m)}(x, \mu_1)$ and $G_2^{(m)}(x, \mu_2)$ in canonical notation as,

$$G_1^{(m)}(x, \mu_1) = \exp \left( -\frac{1}{2} x^T \Sigma_1^{-1} x + (\Sigma_1^{-1} \mu_1)^T x - \frac{1}{2} \mu_1^T \Sigma_1^{-1} \mu_1 \right)$$

$$G_2^{(m)}(x, \mu_2) = \exp \left( -\frac{1}{2} x^T \Sigma_2^{-1} x + (\Sigma_2^{-1} \mu_2)^T x - \frac{1}{2} \mu_2^T \Sigma_2^{-1} \mu_2 \right)$$

Therefore,

$$G_1^{(m)} \cdot G_2^{(m)} = \exp \left( -\frac{1}{2} x^T (\Sigma_1^{-1} + \Sigma_2^{-1}) x + (\Sigma_1^{-1} \mu_1 + \Sigma_2^{-1} \mu_2)^T x - \frac{1}{2} \mu_1^T \Sigma_1^{-1} \mu_1 - \frac{1}{2} \mu_2^T \Sigma_2^{-1} \mu_2 \right)$$

$$\cdot \exp \left( -\frac{1}{2} (\mu_1^T \Sigma_1^{-1} \mu_1 - \mu_2^T (\Sigma_1^{-1} + \Sigma_2^{-1}) \mu_2) \right)$$

where

$$\mu^* = (\Sigma_1^{-1} + \Sigma_2^{-1})^{-1} (\Sigma_1^{-1} \mu_1 + \Sigma_2^{-1} \mu_2)$$

$$= \Sigma_1 (\Sigma_1 + \Sigma_2)^{-1} \mu_1 + \Sigma_2 (\Sigma_1 + \Sigma_2)^{-1} \mu_2.$$

Then, we have

$$G_1^{(m)} \cdot G_2^{(m)} = \exp \left( -\frac{1}{2} (x - \mu^*)^T (\Sigma_1^{-1} + \Sigma_2^{-1}) (x - \mu^*) \right)$$

$$\cdot \exp \left( -\frac{1}{2} (\mu_1 - \mu_2)^T (\Sigma_1 + \Sigma_2)^{-1} (\mu_1 - \mu_2) \right).$$

According to the Gaussian integral

$$\int_{\mathbb{R}^n} \exp \left( -\frac{1}{2} x^T \Sigma x \right) dx = \sqrt{\frac{(2\pi)^n}{|\Sigma|}},$$

we have

$$MKC(\mu_1, \mu_2) = \exp \left( -\frac{1}{2} (\mu_1 - \mu_2)^T (\Sigma_1 + \Sigma_2)^{-1} (\mu_1 - \mu_2) \right).$$
we finally have the Gaussian kernel correlation as,

$$MKC(\mu_1, \mu_2) = \frac{1}{\sqrt{2\pi^{n}}} \exp \left( -\frac{1}{2} \sum_{i=1}^{n} \left( x_i - \mu_i \right)^2 \right).$$

REFERENCES


